Temporal flows in Temporal networks ⋆, ⋆⋆

Eleni C. Akrida[1], Jurek Czyzowicz[2], Leszek Gąsieniec[1],

Łukasz Kuszner[3], and Paul G. Spirakis[1,4]

[1] Department of Computer Science, University of Liverpool, UK
{Eleni.Akrida,L.A.Gasieniec,P.Spirakis}@liverpool.ac.uk
[2] Université du Québec en Outaouais, Dep. d'Informatique, Gatineau, QC, Canada
jurek@uqo.ca
[3] Gdańsk University of Technology, Faculty of Electronics, Telecommunications and Informatics, Poland
kuszner@eti.pg.gda.pl
[4] Computer Technology Institute & Press "Diophantus" (CTI), Patras, Greece

Abstract. We introduce temporal flows on temporal networks [17, 19], i.e., networks the links of which exist only at certain moments of time. Such networks are ephemeral in the sense that no link exists after some time. Our flow model is new and differs from the "flows over time" model, also called "dynamic flows" in the literature. We show that the problem of finding the maximum amount of flow that can pass from a source vertex s to a sink vertex t up to a given time is solvable in Polynomial time, even when node buffers are bounded. We then examine mainly the case of unbounded node buffers. We provide a simplified static *Time-Extended network* (STEG), which is of *polynomial size to the input* and whose static flow rates are equivalent to the respective temporal flow of the temporal network; using STEG, we prove that the maximum temporal flow is equal to the minimum *temporal s-t cut*. We further show that temporal flows can always be decomposed into flows, each of which moves only through a journey, i.e., a directed path whose successive edges have strictly increasing moments of existence. We partially characterise networks with random edge availabilities that tend to eliminate the $s \to t$ temporal flow. We then consider *mixed* temporal networks, which have some edges with specified availabilities and some edges with random availabilities; we show that it is $\#\mathbf{P}$-hard to compute the *tails and expectations of the maximum temporal flow* (which is now a random variable) in a mixed temporal network.

⋆ This work was partially supported by (i) the School of EEE and CS and the NeST initiative of the University of Liverpool, (ii) the NSERC Discovery grant, (iii) the Polish National Science Center grant DEC-2011/02/A/ST6/00201, and (iv) the FET EU IP Project MULTIPLEX under contract No. 317532.

⋆⋆ Due to lack of space, an extended literature review and all missing proofs can be found in the full version of this paper at http://arxiv.org/abs/1606.01091 [2]

1 Introduction and motivation

1.1 Our model, the problem, and our results

It is generally accepted to describe a network topology using a graph, whose vertices represent the communicating entities and edges correspond to the communication opportunities between them. Consider a directed graph (network) $G(V, E)$ with a set V of n vertices (nodes) and a set E of m edges (links). Let $s, t \in V$ be two special vertices called the *source* and the *sink*, respectively; for simplicity, assume that no edge enters the source s and no edge leaves the sink t. We also assume that an infinite amount of a quantity, say, a liquid, is available in s at time zero. However, our network is *ephemeral*; each edge is available for use only at certain *days* in time, described by positive integers, and after some (finite) day in time, no edge becomes available again ; the reader may think of these days as instances of availability of that edge. Our liquid, located initially at node s, can flow in this ephemeral network through edges only at days at which the edges are available.

Each edge $e \in E$ in the network is also equipped with a *capacity* $c_e > 0$ which is a positive integer, unless otherwise specified. We also consider each node $v \in V$ to have an internal buffer (storage) $B(v)$ of maximum size B_v; here, B_v is also a positive integer; initially, we shall consider both the case where $B_v = +\infty$, for all $v \in V$, and the case where all nodes have finite buffers. From Section 3 on, we only consider unbounded (infinite) buffers.

The *semantics* of the flow of our liquid within G are the following:

- Let an amount x_v of liquid be at node v, i.e., in $B(v)$, at the *beginning* of day l, for some $l \in \mathbb{N}$. Let $e = (v, w)$ be an edge that exists at day l. Then, v may *push* some of the amount x_v through e at day l, as long as that amount is at most c_e. This quantity will arrive to w at the *end* of *the same day*, l, and will be stored in $B(w)$.
- At the end of day l, for any node w, some flows may arrive from edges (v, w) that were available at day l. Since each such quantity of liquid has to be stored in w, the sum of all flows incoming to w plus the amount of liquid that is already in w at the end of day l, after w has sent any flow out of it at the beginning of day l, must not exceed B_w.
- Flow arriving at w at (the end of) day l can leave w only via edges existing at days $l' > l$.

Thus, our flows are not flow rates, but flow amounts (similar to considerations in *transshipment problems* [14, 16]). Notice that we assume above that we have absolute knowledge of the days of existence of each edge. Admittedly, the encoding of the input in our temporal network problems is quite detailed but specific description of the edge availabilities (or lack thereof) may be required in a range of network infrastructure settings where there is a planned schedule of link existence, e.g., one may need to have detailed information on planned maintenance on pipe-sections in a water network to assure restoration of the network services. On the positive side, some problems that are weakly NP-hard in similar dynamic flow models become polynomially solvable in our model.

Our results. We provide polynomial-time solutions to the *Maximum temporal flow problem* (MTF): Given a directed graph G with edge availabilities, distinguished nodes s, t, edge capacities and node buffers as previously described, and also given a specific day $l' > 0$, find the maximum value of the quantity of liquid that can arrive to t by (the end of) day l'.

For the case of infinite buffers, we give a simplified static *Time-Extended network* (STEG) which, in contrast to all previous dynamic flows literature and due to the encoding of our input, is of *linear* size to the input, and *not exponential*. The static flow rates of STEG are equivalent to the respective temporal flow of the temporal network; using it, we prove that the maximum temporal flow is equal to the minimum *temporal s-t cut*. We also show that temporal flows can always be decomposed into flows, each of which moves only through a journey, i.e., a directed path whose successive edges have strictly increasing moments of existence.

In many practical scenarios it is reasonable to assume that not all edge availabilities are known in advance, e.g., in a water network where there may be unplanned disruptions at one or more pipe sections; in these cases, one may have statistical information on the pattern of link availabilities. We partially characterise networks with random edge availabilities that tend to eliminate the $s \to t$ temporal flow. We also introduce and study here flows in *mixed temporal* networks for the first time; these are networks in which the availabilities of some edges are random and the availabilities of some other edges are specified. In such networks, the value of the maximum temporal flow is a random variable. Consider, for example, the temporal flow network of Figure 1 where there are n directed disjoint two-edge paths from s to t. Assume that *every* edge independently selects a *single* label uniformly at random from the set $\{1, \ldots, \alpha\}$, $\alpha \in \mathbb{N}^*$. The edge capacities are the numbers drawn in the boxes, with $w'_i \geq w_i$ for all i. Here, the value of the maximum $s \to t$ flow is a random variable that is the sum of Bernoulli random variables. This already indicates that the exact calculation of the maximum flow in mixed networks is a hard problem; we show for mixed networks that it is **#P**-hard to compute tails and expectations of the maximum temporal flow.

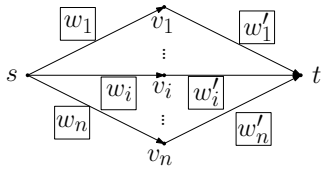

Fig. 1. A mixed temporal network

1.2 Previous work

The traditional (static) network flows were extensively studied in the seminal book of Ford and Fulkerson [13] (see also Ahuja et al [1]) and the relevant literature is vast. They have recently been re-examined for the purpose of approximating their maximum value or improving their time complexity [8,18,20,23,24]. *Dynamic network flows* (also called *flows over time*) [15] refer to *static* directed networks, the edges of which have capacities as well as transit times. Ford and Fulkerson [13] formulated and solved the dynamic maximum flow problem. For excellent surveys on dynamic network flows, the reader is also referred to the work of Aronson [6], the work of Powell [22], and the great survey by Skutella [25].

Temporal networks, defined by Kempe et al. [17], are graphs *the edges of which exist only at certain instants of time, called labels* (see also [19]). So, they are a type of *dynamic* networks. Various aspects of temporal (and other dynamic) networks were also considered in the work of Erlebach et al [12] and in [4,5,7,9]; as far as we know, this is the first work to examine flows on temporal networks. There is also literature on models of temporal networks with random edge availabilities [3, 10, 11], but to the best of our knowledge, ours is the first work on flows in such temporal networks.

Perhaps the closest model in the flows literature to our model is the "*Dynamic*[5] dynamic network flows", studied by Hoppe in his PhD thesis [15, Chapter 8]. Hoppe introduces *mortal edges* that exist between a start and an end time; still, Hoppe assumes transmission rates on the edges and the ability to hold any amount of flow on a node (infinite node buffers). Thus, our model is an extreme case of the latter, since we assume that edges exist only at specific days (instants) and that our transit rates are virtually unbounded, since at one instant *any amount* of flow can be sent through an edge if the capacity allows.

1.3 Formal Definitions

Definition 1 ((Directed) Temporal Graph). *Let $G = (V, E)$ be a directed graph. A (directed) temporal graph on G is an ordered triple $G(L) = (V, E, L)$, where $L = \{L_e \subseteq \mathbb{N} : e \in E\}$ assigns a* finite *set L_e of discrete labels to every edge (arc) e of G. L is called the* labelling *of G. The labels, L_e, of an edge $e \in E$ are the* integer time instances (e.g., days) *at which e is available.*

Definition 2 (Time edge). *Let $e = (u, v)$ be an edge of the underlying digraph of a temporal graph and consider a label $l \in L_e$. The ordered triplet (u, v, l), also denoted as (e, l), is called* time edge. *We denote the set of time edges of a temporal graph $G(L)$ by E_L.*

A basic assumption that we follow is that when a (flow) entity passes through an available edge e at time t, then it can pass through a subsequent edge only at some time $t' \geq t + 1$ and only at a time at which that edge is available. In the tradition of assigning "transit times" in the dynamic flows literature,

[5] The first "dynamic" term refers to the dynamic nature of the underlying graph

one may think that any edge e of the graph has some *transit time*, tt_e, with $0 < tt_e < 1$, but *otherwise arbitrary and not specified*. Henceforth, we assume $tt_e = 0.5,\ \forall e \in E$, without loss of generality; any value of tt_e between 0 and 1 will lead to the same results in our paper.

Definition 3 (Journey). *A* journey *from a vertex u to a vertex v ($u \to v$* journey*) is a sequence of time edges (u, u_1, l_1), ... , (u_{k-1}, v, l_k), such that $l_i < l_{i+1},\ \forall i = 1, \ldots, k-1$. The last label, l_k, is called the* arrival time *of the journey.*

Definition 4 (Foremost journey). *A $u \to v$ journey in a temporal graph is called* foremost journey *if its arrival time is the minimum arrival time of all $u \to v$ journeys' arrival times, under the labels assigned to the underlying graph's edges. We call this arrival time the* temporal distance*, $\delta(u, v)$, of v from u.*

Thus, no flow arrives to t (starting from s) on or before any time $l < \delta(s, t)$.

Definition 5 (Temporal Flow Network). *A temporal flow network $\big(G(L), s, t, c, B\big)$ is a temporal graph $G(L) = (V, E, L)$ equipped with:*

1. *a source vertex s and a sink (target) vertex t*
2. *for each edge e, a capacity $c_e > 0$; usually the capacities are assumed to be integers.*
3. *for each node v, a buffer $B(v)$ of storage capacity $B_v > 0$; we assume $B_s = B_t = +\infty$.*

If all node capacities are infinite, we denote the network by $\big(G(L), s, t, c\big)$.

Definition 6 (Temporal Flows in Temporal Flow Networks). *Let $\big(G(L) = (V, E, L), s, t, c, B\big)$ be a temporal flow network. Denote by δ_u^+ the outgoing edges from u and by δ_u^- the incoming edges to u. Let $L_R(u)$ be the set of labels on all edges incident to u along with an extra label 0 (artificial label for initialization), i.e., $L_R(u) = \bigcup_{e \in \delta_u^+ \cup \delta_u^-} L_e \cup \{0\}$. A* temporal flow *on $G(L)$ consists of a non-negative real number $f(e, l)$ for each time-edge (e, l), and real numbers $b_u^-(l), b_u^\mu(l), b_u^+(l)$ for each node $u \in V$ and each "day" l, such that:*

1. *$0 \le f(e, l) \le c_e$, for every time edge (e, l),*
2. *$0 \le b_u^-(l) \le B_u,\ 0 \le b_u^\mu(l) \le B_u,\ 0 \le b_u^+(l) \le B_u$, for every node u and every $l \in L_R(u)$*
3. *for every $e \in E$, $f(e, 0) = 0$,*
4. *for every $v \in V \setminus \{s\}$, $b_v^-(0) = b_v^\mu(0) = b_v^+(0) = 0$,*
5. *for every $e \in E$ and $l \notin L_e$, $f(e, l) = 0$,*
6. *at time 0 there is an infinite amount of flow "units" available at the source s,*
7. *for every $v \in V \setminus \{s\}$ and for every $l \in L$, $b_v^-(l) = b_v^+(l_{prev})$, where l_{prev} is the largest label in $L_R(v)$ that is smaller than l,*
8. *(Flow out on day l) for every $v \in V \setminus \{s\}$ and for every l, $b_v^\mu(l) = b_v^-(l) - \sum_{e \in \delta_v^+} f(e, l)$,*
9. *(Flow in on day l) for every $v \in V \setminus \{s\}$ and for every l, $b_v^+(l) = b_v^\mu(l) + \sum_{e \in \delta_v^-} f(e, l)$.*

Note 1 *One may think of $b_v^-(l), b_v^\mu(l), b_v^+(l)$ as the buffer content of liquid in v at the "morning","noon", i.e., after the departures of flow from v, and "evening", i.e., after the arrivals of flow to v, of day l.*

Note 2 *For a temporal flow f on an acyclic $G(L)$, if one could guess the (real) numbers $f(e,l)$ for each time-edge (e,l), then the numbers $b_v^-(l), b_v^\mu(l), b_v^+(l)$, for every $v \in V$, can be computed by a single pass over an order of the vertices of $G(L)$ from s to t. This can be done by following (1) through (9) from Definition 6 from s to t.*

Definition 7 (Value of a Temporal Flow). *The value $v(f)$ of a temporal flow f is $b_t^+(l_{max})$ under f, i.e., the amount of liquid that, via f, reaches t during the lifetime of the network (l_{max} is the maximum label in L). If $b_t^+(l_{max}) > 0$ for a particular flow f, we say that f is* feasible.

Definition 8 (Mixed temporal networks). *Given a directed graph $G = (V, E)$ with a source s and a sink t in V, let $E = E_1 \cup E_2$, so that $E_1 \cap E_2 = \emptyset$, and:*

1. *the labels (availabilities) of edges in E_1 are specified, and*
2. *each of the labels of the edges in E_2 is drawn uniformly at random from the set $\{1, 2, \ldots, \alpha\}$, for some even integer α[6], independently of the others.*

We call such a network "Mixed Temporal Network $[1, \alpha]$" *and denote it by $G(E_1, E_2, \alpha)$.*

Note that (traditional) temporal networks as previously defined are a special case of the mixed temporal networks, in which $E_2 = \emptyset$. However, with some edges being available at random times, the value of a temporal flow (until time α) becomes a random variable and the study of relevant problems requires a different approach than the one needed for (traditional) temporal networks.

Problem 1 (Maximum Temporal Flow (MTF)) *Given a temporal flow network $\big(G(L), s, t, c, B\big)$ and a day $d \in \mathbb{N}^*$, compute the maximum $b_t^+(d)$ over all flows f in the network.*

2 LP for the MTF problem with or without bounded buffers

In the description of the MTF problem, if d is not a label in L, it is enough to compute the maximum $b_t^+(l_m)$ over all flows, where l_m is the maximum label in L that is smaller than d. Henceforth, we assume $d = l_{max}$ unless otherwise specified; the analysis does not change: if $d < l_{max}$, one can remove all time-edges with labels larger than b and solve MTF in the resulting network.

[6] We choose an even integer to simplify the calculations in the remainder of the paper. However, with careful adjustments, the results would still hold for an arbitrary integer.

Let Σ be the set of conditions of Definition 6. The optimization problem, Π:

$$\left\{ \begin{array}{ll} \max \text{ (over all } f) & b_t^+(d) \\ \text{subject to} & \Sigma \end{array} \right\}$$

is a *linear program* with unknown variables $\{f(e,l), b_v^-(l), b_v^+(l)\}$, $\forall l \in L, \forall v \in V$, since each condition in Σ is either a linear equation or a linear inequality in the unknown variables. Therefore, by noticing that the number of equations and inequalities are polynomial in the size of the input of Π, we get the following:

Lemma 1. *Maximum Temporal Flow is in P, i.e., can be solved in polynomial time in the size of the input, even when the node buffers are finite, i.e., bounded.*

Note 3 *Recall that E_L is the set of time edges of a temporal graph. If $n = |V|, m = |E|$ and $k = |E_L| = \sum_e |L_e|$, then MTF can be solved in sequential time polynomial in $n + m + k$ when the capacities and buffer sizes can be represented with polynomial in n number of bits. In the remainder of the paper, we shall investigate more efficient approaches for MTF.*

3 Temporal Networks with unbounded buffers at nodes

3.1 Basic remarks

We consider here the MTF problem for temporal networks on underlying graphs with $B_v = +\infty, \ \forall v \in V$.

Definition 9 (Temporal Cut). *Let $\big(G(L), s, t, c\big)$ be a temporal flow network on a digraph G. A set of time-edges, S, is called a temporal cut (separating s and t) if the removal from the network of S results in a temporal flow network with* no $s \to t$ journey*.*

Definition 10 (Minimal Temporal Cut). *A set of time-edges, S, is called a minimal temporal cut (separating s and t) if S is a temporal cut, and no proper subset of S is a temporal cut.*

Definition 11. *Let S be a temporal cut of $\big(G(L) = (V, E, L), s, t, c\big)$. The* capacity *of the cut is $c(S) := \sum_{(e,l)\in S} c(e,l)$, where $c(e,l) = c_e, \ \forall l$.*

Lemma 2. *Let S be a (minimal) temporal cut in $\big(G(L) = (V, E, L), s, t, c\big)$. If we remove S from $G(L)$, no flow can ever arrive to t during the lifetime of $G(L)$.*

3.2 The time-extended flow network and its simplification

Let $\big(G(L) = (V, E, L), s, t, c\big)$ be a temporal flow network on a directed graph G. Let E_L be the set of time edges of $G(L)$. Following the tradition in literature [13], we construct the *time-extended* static flow network that corresponds to $G(L)$, denoted by $\text{TEG}(L) = (V^*, E^*)$. By construction, $\text{TEG}(L)$ admits the same

maximum flow as $G(L)$. TEG(L) is constructed as follows: for every vertex $v \in V$ and for every time step $i = 0, 1, \ldots, l_{max}$, V^* contains a copy, v_i, of v. Also, for every time edge (x, v, l), $l \in \mathbb{N}$, $x \in V$ of $G(L)$, V^* contains a copy $v_{l+0.5}$ of v. E^* has a directed edge (called *vertical*) from a copy of vertex v to the *next* copy of v, for any $v \in V$, where the order of the copies is defined by their indices; every vertical edge has infinite capacity (as the node whose copies it connects). Furthermore, for every time edge (u, v, l) of $G(L)$, E^* has a directed edge (called *crossing*) $(u_l, v_{l+0.5})$ with capacity equal to the capacity of the edge (u, v). The source and target vertices in TEG(L) are the first copy of s and the last copy of t in V^*, respectively. Note that $|V^*| \leq |V| \cdot l_{max} + |E_L|$ and $|E^*| \leq |V| \cdot l_{max} + 2|E_L|$.

We now "simplify" TEG(L) as follows: we convert vertical edges between consecutive copies of the same vertex into a *single vertical edge (with infinite capacity)* from the first to the last copy in the sequence and we remove all intermediate copies; we only perform this simplification when no intermediate node is an endpoint of a crossing edge. We call the resulting network *simplified time-extended* network and we denote it by STEG$(L) = (V', E')$. Note that $|V'| \leq |V| + 2|E_L|$ and $|E'| \leq |V| + 3|E_L|$.

Let the first copy of any vertex $v \in V$ in the time-extended network be v_{copy_0}, the second copy v_{copy_1}, etc. An $s \to t$ flow f in $G(L)$ defines an $s \to t$ flow in the time-extended network STEG(L) as follows:

- The flow from the first copy of s to the next copy is the sum of all flow units that "leave" s in $G(L)$ throughout the time the network exists.
- The flow from the first copy of any *other* vertex to the next copy is zero.
- The flow on any crossing edge that connects some copy u_l of vertex $u \in V$ and the copy $v_{l+0.5}$ of some other vertex $v \in V$ is exactly the flow on the time edge (u, v, l).
- The flow between two *consecutive* copies v_x and v_y, for some x, y, of the same vertex $v \in V$ corresponds to the units of flow stored in v from time x up to time y and is the difference between the flow *received* at the first copy through all incoming edges and the flow *sent* from the first copy through all outgoing *crossing* edges.

Using TEG(L) and STEG(L), we can prove the following (for the proof, see [2]):

Theorem 1. *The maximum temporal flow in* $\big(G(L) = (V, E, L), s, t, c\big)$ *is equal to the minimum capacity (minimal) temporal cut.*

Lemma 3. *Any static flow rate algorithm A that computes the maximum flow in a static, directed, s-t network G of n vertices and m edges in time* $T(n, m)$*, also computes the maximum temporal flow in a* $\big(G(L) = (V, E, L), s, t, c\big)$ *temporal flow network in time* $T(n', m')$*, where* $n' \leq n + 2|E_L|$ *and* $m' \leq n + 3|E_L|$.

Corollary 1 (Journeys flow decomposition). *Let* $\big(G(L) = (V, E, L), s, t, c\big)$ *be a temporal flow network on a directed graph G. Let f be a temporal flow in* $G(L)$ *(f is given by the values of* $f(e, l)$ *for the time-edges* $(e, l) \in E_L$*). Then, there is a collection of* $s \to t$ *journeys* $j_1, j_2, \ldots, j_k$ *such that:*

1. $k \leq |E_L|$
2. $v(f) = v(f_1) + \ldots v(f_k)$
3. f_i *sends positive flow only on the time-edges of* j_i

4 Mixed Temporal Networks and their hardness

Mixed temporal networks of the form $G(E_1, E_2, \alpha)$ (see Definition 8) can model practical cases, where some edge availabilities are exactly specified, while some other edge availabilities are randomly chosen (due to security reasons, faults, etc.); for example, in a water network, one may have planned disruptions for maintenance in some water pipes, but unplanned (random) disruptions in some others. With some edges being available at random times, the value of the maximum temporal flow (until time α) now becomes a random variable.

4.1 Temporal Networks with random availabilities that are flow cutters

We study here a special case of the mixed temporal networks $G(E_1, E_2, \alpha)$, where $E_1 = \emptyset$, i.e., *all* edges become available at random time instances, and we partially characterise such networks that eliminate the flow that arrives at t asymptotically almost surely. All missing proofs can be found in the full version of the paper [2].

Let $G = (V, E)$ be a directed graph of n vertices with a distinguished source, s, and a distinguished sink, t. Suppose that each edge $e \in E$ is available only at a *unique* moment in time (i.e., day) *selected uniformly at random from the set* $\{1, 2, \ldots, \alpha\}$*, for some even*[7] *integer* $\alpha \in \mathbb{N},\ \alpha > 1$; suppose also that the selections of the edges' labels are independent. Let us call such a network a Temporal Network with unique random availabilities of edges, and denote it by URTN(α). Then, the following holds:

Lemma 4. *Let P_k be a directed $s \to t$ path of length k in G. Then, P_k becomes a journey in* URTN(α) *with probability at most $\frac{1}{k!}$.*

Now, consider directed graphs as described above, in which the distance from s to t is at least $c \log n$, for a constant integer $c > 2$; so any directed $s \to t$ path has at least $c \log n$ edges. Let us call such graphs " c-long $s \to t$ graphs" or simply c-long. A c-long $s \to t$ graph is called *thin* if the number of simple directed $s \to t$ paths is at most n^β, for some constant β. It can be proven that:

Lemma 5. *Consider a $URTN(\alpha)$ with an underlying graph G being any particular c-long and thin digraph. Then, the probability that the amount of flow from s arriving at t is positive tends to zero as n tends to $+\infty$.*

Randomly labelled c-long and thin graphs is not the only case of temporal networks that disallows flow to arrive to t asymptotically almost surely.

Definition 12. *A cut C in a (traditional) flow network G is a set of edges, the removal of which from the network leaves no directed $s \to t$ paths in G.*

[7] We choose an even integer to simplify the calculations. However, with careful adjustments in the calculations, the results would still hold for an arbitrary integer.

Definition 13. *A cut* C_1 precedes *a cut* C_2 *in a flow network* G *(denoted by* $C_1 \to C_2$*) if any directed* $s \to t$ *path that goes through an edge in* C_1 *must also later go through an edge in* C_2*.*

Definition 14 (Multiblock graphs). *A flow network is called a* (c, d)*-multiblock graph if it has at least* $c \log n$ *disjoint cuts* $C_1, \ldots, C_{c \log n}$ *such that* $C_i \to C_{i+1},\ i = 1, \ldots, c \log n - 1$*, and for all* $i = 1, \ldots, c \log n$*,* $|C_i| \leq d$*, for some constants* $c > 2,\ d \geq 2$*.*

Note that (c, d)-multiblocks and (c-long,thin)-graphs are two different graph classes. Figure 2 shows a $(c, 2)$-multiblock of $n = c\sqrt{k} + 2,\ k \in \mathbb{N}$, vertices which is not thin.

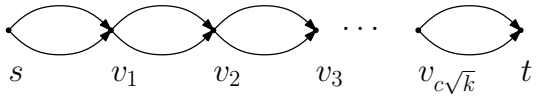

Fig. 2. A $(c, 2)$-multiblock which is not thin.

Lemma 6. *Consider a* $URTN(\alpha)$ *with an underlying graph* G *being any particular* (c, d)*-multiblock. Then, the probability that the amount of flow from* s *arriving at* t *is positive tends to zero as* n *tends to* $+\infty$*.*

4.2 The complexity of computing the expected maximum temporal flow

We consider here the following problem:

Problem 2 (Expected Maximum Temporal Flow) *What is the time complexity of computing the* expected value *of the maximum temporal flow,* v*, in* $G(E_1, E_2, \alpha)$*?*

Definition 15. *[21, p.441] Let* Q *be a polynomially balanced, polynomial-time decidable binary relation. The* counting problem *associated with* Q *is: Given* x*, how many* y *are there such that* $(x, y) \in Q$*?* ***#P*** *is the class of all counting problems associated with polynomially balanced polynomial-time decidable functions.*

Loosely speaking, a problem is said to be **#P**-hard if a polynomial-time algorithm for it implies that $\#\mathbf{P} = \overline{\mathbf{FP}}$, where $\overline{\mathbf{FP}}$ is the set of functions from $\{0,1\}^*$ to $\{0,1\}^*$ computable by a deterministic polynomial-time Turing machine[8]. For a more formal definition, see [21]. We show the following:

Lemma 7. *Given an integer* $C > 0$*, it is* ***#P****-hard to compute the probability that the maximum flow value* v *in* $G(E_1, E_2, \alpha)$ *is at most* C*,* $Pr[v \leq C]$*.*

[8] $\{0,1\}^* = \cup_{n \geq 0}\{0,1\}^n$, where $\{0,1\}^n$ is the set of all strings (of bits $0, 1$) of length n

Now, given a mixed temporal network $G(E_1, E_2, \alpha)$, let v be the random variable representing the maximum temporal flow in G.

Definition 16. *The truncated by B expected maximum temporal flow of $G(E_1, E_2, \alpha)$, denoted by $E[v, B]$, is defined as: $E[v, B] = \sum_{i=1}^{B} iPr[v = i]$. Clearly, it is $E[v] = E[v, +\infty]$.*

The following is the main theorem of this section.

Theorem 2. *It is **#P**-hard to compute the expected maximum truncated Temporal Flow in a Mixed Temporal Network $G(E_1, E_2, \alpha)$.*

Open Problem 1 *Is there an FPTAS for the expected maximum flow value in mixed temporal networks?*

Open Problem 2 *What is the complexity of the maximum flow problem in* periodic *temporal graphs? These are graphs each edge e of which appears every x_e days ("edge period"). The maximum flow from s to t would then, in general, increase as we increase the day by which we wish to compute the flow that arrives at t. It seems that this problem requires a different approach than the one presented here, that also takes into account the different edge periods.*

References

1. R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.
2. E. C. Akrida, J. Czyzowicz, L. Gasieniec, L. Kuszner, and P. G. Spirakis. Flows in temporal networks. *CoRR*, abs/1606.01091, 2016.
3. E. C. Akrida, L. Gasieniec, G. B. Mertzios, and P. G. Spirakis. Ephemeral networks with random availability of links: The case of fast networks. *J. Parallel Distrib. Comput.*, 87:109–120, 2016.
4. E. C. Akrida, L. Gąsieniec, G. B. Mertzios, and P. G. Spirakis. On temporally connected graphs of small cost. In *Proceedings of the 13th Workshop on Approximation and Online Algorithms (WAOA)*, 2015.
5. E. C. Akrida and P. G. Spirakis. On verifying and maintaining connectivity of interval temporal networks. In *Algorithms for Sensor Systems - 11th International Symposium on Algorithms and Experiments for Wireless Sensor Networks, ALGOSENSORS 2015, Patras, Greece, September 17-18, 2015, Revised Selected Papers*, pages 142–154, 2015.
6. J. E. Aronson. A survey of dynamic network flows. *Ann. Oper. Res.*, 20(1-4):1–66, Aug. 1989.
7. C. Avin, M. Koucký, and Z. Lotker. How to explore a fast-changing world (cover time of a simple random walk on evolving graphs). In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 121–132, 2008.
8. J. Batra, N. Garg, A. Kumar, T. Mömke, and A. Wiese. New approximation schemes for unsplittable flow on a path. In P. Indyk, editor, *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 47–58. SIAM, 2015.

9. A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro. Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems (IJPEDS)*, 27(5):387–408, 2012.
10. A. Chaintreau, A. Mtibaa, L. Massoulié, and C. Diot. The diameter of opportunistic mobile networks. In *Proceedings of the 2007 ACM Conference on Emerging Network Experiment and Technology, CoNEXT 2007, New York, NY, USA, December 10-13, 2007*, page 12, 2007.
11. A. E. F. Clementi, C. Macci, A. Monti, F. Pasquale, and R. Silvestri. Flooding time of edge-markovian evolving graphs. *SIAM Journal on Discrete Mathematics (SIDMA)*, 24(4):1694–1712, 2010.
12. T. Erlebach, M. Hoffmann, and F. Kammer. On temporal graph exploration. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, pages 444–455, 2015.
13. D. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, NJ, USA, 2010.
14. B. Hoppe and E. Tardos. The quickest transshipment problem. In *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '95, pages 512–521, Philadelphia, PA, USA, 1995. Society for Industrial and Applied Mathematics.
15. B. E. Hoppe. Phd thesis: Efficient dynamic network flow algorithms, 1995.
16. N. Kamiyama and N. Katoh. The universally quickest transshipment problem in a certain class of dynamic networks with uniform path-lengths. *Discrete Applied Mathematics*, 178:89–100, 2014.
17. D. Kempe, J. M. Kleinberg, and A. Kumar. Connectivity and inference problems for temporal networks. In *Proceedings of the 32nd annual ACM symposium on Theory of computing (STOC)*, pages 504–513, 2000.
18. A. Madry. Fast approximation algorithms for cut-based problems in undirected graphs. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 245–254, 2010.
19. G. B. Mertzios, O. Michail, I. Chatzigiannakis, and P. G. Spirakis. Temporal network optimization subject to connectivity constraints. In *Proceedings of the 40th International Colloquium on Automata, Languages and Programming (ICALP), Part II*, pages 657–668, 2013.
20. J. B. Orlin. Max flows in o(nm) time, or better. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 765–774, 2013.
21. C. M. Papadimitriou. *Computational complexity*. Addison-Wesley, Reading, Massachusetts, 1994.
22. W. B. Powell, P. Jaillet, and A. Odoni. Stochastic and dynamic networks and routing. *Handbooks in operations research and management science*, 8:141–295, 1995.
23. T. Radzik. Faster algorithms for the generalized network flow problem. *Mathematics of Operations Research*, 23(1):69–100, 1998.
24. M. J. Serna. Randomized parallel approximations to max flow. In *Encyclopedia of Algorithms*, pages 1750–1753. 2016.
25. M. Skutella. An introduction to network flows over time. In *Research Trends in Combinatorial Optimization, Bonn Workshop on Combinatorial Optimization, November 3-7, 2008, Bonn, Germany*, pages 451–482, 2008.